

Quarkus Grundlagen

JEE Microservices

@ CGS IT – 2023

Version 1.0.5



QUARKUS

Inhalt

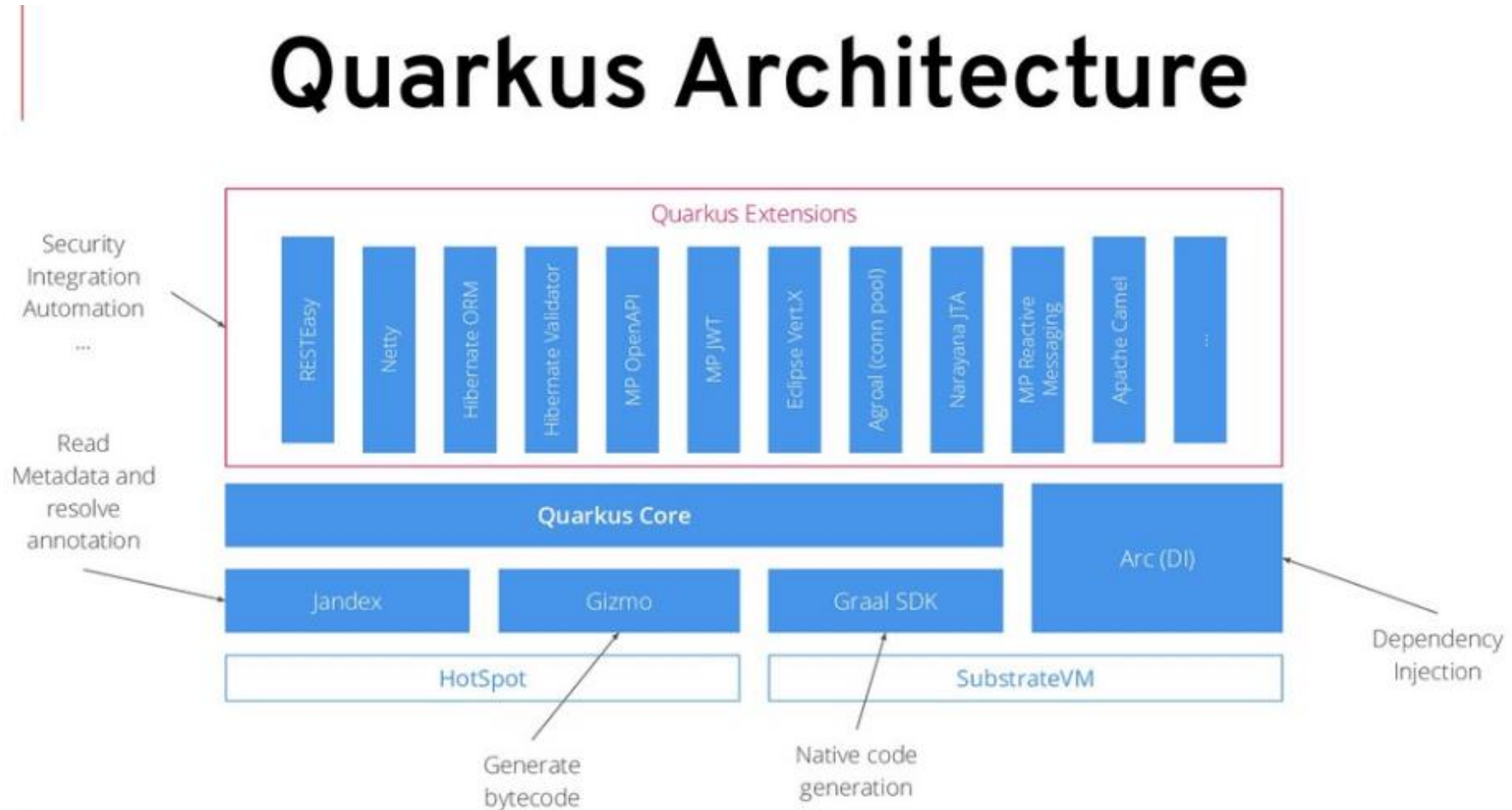
- Was ist Quarkus?
- Quarkus - Architektur
- GraalVM
- Quarkus Vorteile
- Quarkus Container CDI
- Quarkus – Spring DI Compatibility
- Quarkus CDI Limitations
- Quarkus Installation Requirements
- Quarkus Installation -

Was ist Quarkus?

*Supersonic, Subatomic,
Hot Reloading und Container First*

- Quarkus ist ein Framework, das auf den besten Java-Bibliotheken und -Mustern basiert.
- Das Hauptziel von Quarkus ist es, eine schnellere Entwicklung und Anwendungsinitialisierung zu ermöglichen, kleiner zu sein und weniger Maschinenressourcen zu verbrauchen.
- Quarkus wurde mit der Idee geboren, Cloud Native zu sein und in Microservices- und Serverless-Projekten verwendet zu werden.
- seit 12/12/2018 (RedHat)
- lässt sich die darauf aufbauende Entwicklung „GraalVM“

Quarkus - Architektur



Quarkus Extensions

From inside a Quarkus project, you can obtain a list of the available extensions with:

Dealing with extensions

From inside a Quarkus project, you can obtain a list of the available extensions with:

```
./mvnw quarkus:list-extensions
```

You can enable an extension using:

```
./mvnw quarkus:add-extension -Dextensions="hibernate-validator"
```

Extensions are passed using a comma-separated list.

GraalVM

- erlaubt das **Kompilieren des Java-Codes in direkt ausführbaren Maschinencode**
- **Performance-Vorteile Quarkus & GraalVM:**
 - Schnelle Startzeit der Anwendung
 - Geringer Speicherverbrauch der laufenden Anwendung
 - Beinahe unmittelbare Skalierung von Services
 - Geringer Platzbedarf der nativen Images

Quarkus Vorteile

Standard-Technologien:

1. Eclipse MicroProfile
2. Dependency Injection Support
3. Hibernate ORM
4. Etc.

Quarkus Container CDI

- Dependency injection in Quarkus is based on ArC which is a CDI-based dependency injection solution tailored for Quarkus' architecture.
- If you're new to CDI then we recommend you to read the [Introduction to CDI](#) guide.
- Quarkus only implements a subset of the CDI features and comes with non-standard features and specific APIs,
- you can learn more about it in the [Contexts and Dependency Injection guide](#).

Quarkus – Spring DI Compatibility

- While users are encouraged to use CDI annotations for injection,
- Quarkus provides a compatibility layer for Spring dependency injection
- in the form of the spring-di extension.

If you already have your Quarkus project configured, you can add the `spring-di` extension to your project by running the following command in your project base directory:

```
./mvnw quarkus:add-extension -Dextensions="spring-di"
```

This will add the following to your `pom.xml`:

```
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-spring-di</artifactId>
</dependency>
```

Quarkus Installation Requirements

Quarkus Installation -

1. IntelliJ / Eclipse
2. JDK 11+ (any distribution)
Optionally get GraalVM 21.3.0 for native compilation
3. Apache Maven 3.8.1+ (or Gradle)

Quarkus – Blick über den Tellerrand



(Bild: Shutterstock)

Danke für Ihre Aufmerksamkeit